

**REMARKS**

The non-final Office Action of 16 May 2008 has been received and its contents carefully studied. Reconsideration of the rejections of the claims is respectfully requested in view of the preceding amendments and the following remarks.

The independent claims are claims 1, 11, 19, 22, and 33. All of these independent claims stand rejected as obvious under 35 U.S.C. 103(a) from *Blow* (WO 99/53621).

The non-final Office Action states at page 13: “the claims do not specifically recite the method of making the library available as if the library were installed on the device.” Applicant respectfully submits that this is incorrect. Present claim 1 specifically recites “making said library available...as if said library were installed on said electronic device.” Likewise for the other independent claims, such as method claim 22: “making said library available...as if said library were installed on said electronic device.” Thus, the non-final Office Action is incorrect.

Moreover, the non-final Office Action (at page 13) says that *Blow* discloses “dynamic upload” of interface software. However, even if that were true, a person of ordinary skill understands that downloading or uploading software to an electronic device is unnecessary if software is already installed (or seems to be installed) on said electronic device, as presently claimed.

The *Blow* reference mentions “dynamic” upload at page 2, lines 14-21 and also at page 8, lines 15-24. The latter paragraph makes clear what *Blow* means by “dynamic” upload. *Blow* states at page 8, line 16 that the interface memory must be large enough to store the “entire” accessory interface software for the most complex external accessory anticipated. Thus, *Blow* clearly does not teach or suggest loading only pieces of code as needed from a library to the electronic device, but instead discloses only loading the “entire” interface accessory software when the accessory is used.

Generally speaking, a software library is a collection of more or less related object code. In the Java language, libraries are typically packaged in JAR files, as indicated at page 16 of the present application. Libraries can contain various different sorts of objects, and the most important type of object contained in a JAR file is a Java class. A class can be thought of as a named unit of code. The classloader (mentioned at pages 15 and 16 of the present application) is responsible for locating libraries, reading their contents, and loading the classes contained within the libraries. This loading is typically done "on demand", in that it does not occur until the class is actually used by the program. A class with a given name can only be loaded once by a given classloader.

In order to further expedite prosecution of this application, the independent claims are now amended without prejudice to clarify that the library is not loaded from the accessory device to the electronic device. The amendments introduce no new matter, and are fully supported by the specification as originally filed. *Blow* is entirely different, in that it does load the library from the accessory device to the electronic device, per page 8, lines 15-24 of *Blow*. Also, present claim 37 is further clarified by specifying that classloading occurs from the library to the electronic device.

Applicant respectfully submits that configuring the interface for making the library available to the electronic device during operation of the electronic device directly from the accessory device, without downloading the library from the accessory device, is not obvious to a skilled person familiar with the *Blow* reference.

The *Blow* reference merely discloses "download of the accessory interface software code from accessory interface memory 118 in external accessory 102 to interface upload memory 106 in mobile station 100" (see *Blow* page 6, lines 26-30). In contrast, the present claimed invention is arranged so that no downloading of the entire accessory interface software is needed.

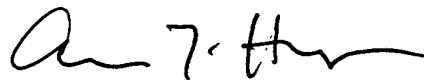
*Blow* teaches that after a successful download of the accessory interface software code from the accessory interface memory of the external accessory to the interface upload

memory in the mobile station, the mobile station controller begins to execute the instructions contained in the accessory interface software. It is therefore clear that the accessory interface software is first entirely downloaded to the mobile station of *Blow*, and then executed from the memory of the mobile station. That software of *Blow* is not accessible as if it were installed on the electronic device; rather, it is downloaded to the electronic device. ***Blow explicitly uses the word "entirely" at page 8, line 16.***

### **CONCLUSION**

The objections and rejections of the Office Action of 16 May 2008 having been obviated by amendment or shown to be inapplicable, withdrawal thereof is requested and passage of claims 1, 2, 4-13, 15-20 and 22-37, as amended, to issue is earnestly solicited. It is requested that the Examiner please contact the undersigned by telephone to set up an appointment for an Examiner's Interview to discuss the foregoing comments, if there is still any doubt about the patentability of the present claims.

Respectfully submitted,



Andrew T. Hyman  
Attorney for the Applicants  
Registration No. 45,858

mbh  
WARE, FRESSOLA, VAN DER SLUYS  
& ADOLPHSON LLP  
755 Main Street, PO Box 224  
Monroe CT 06468  
Tel: (203) 261-1234  
Fax: (203) 261-5676